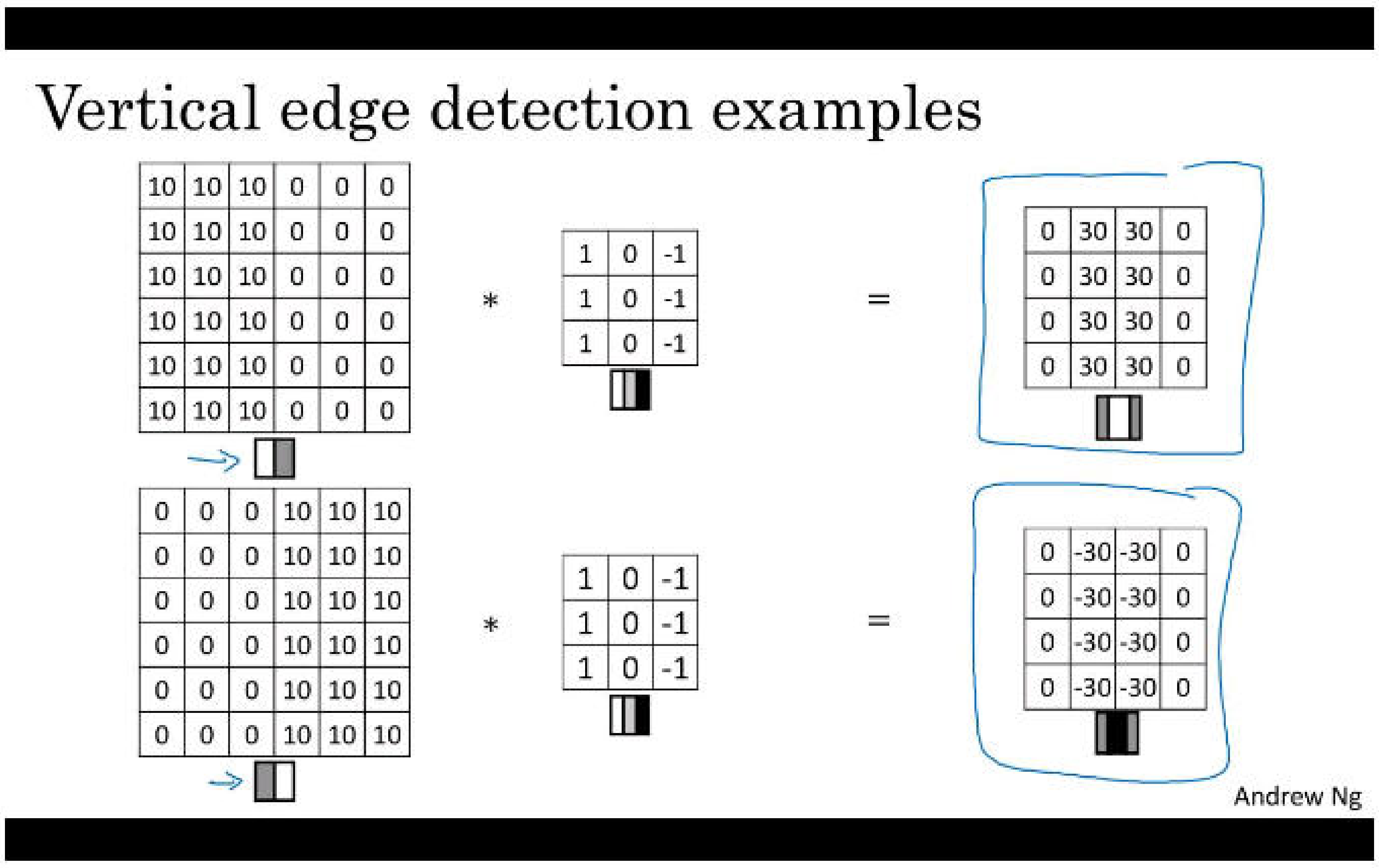
1 11

Deep learning.

Edge detection:



Kernel 能检测出不同性质的垂直边缘.

水平的情况也是类似的.

Padding: $n \times n$ 的图像, $f \times f$ 的卷积核, padding 大小为 p.

卷积后图像大小: $n-f+2p+1$

$$\begin{cases} \text{Valid}: & p = 0, \\ \text{Same}: & n-f+2p+1 = n \quad \Rightarrow \quad p = \frac{f-1}{2}. \end{cases}$$

Stride $= S$, Output size $= \left\lfloor \frac{n+2p-f}{s} \right\rfloor + 1$

卷积与数学上的运算是有差别的,"惯例名称"

n_c 通道做卷积, n_c 层卷积核, 输出一个矩阵.

多个卷积核则输出多个矩阵.

CNN 通常最后一层是 Logistic / SoftMax 层.

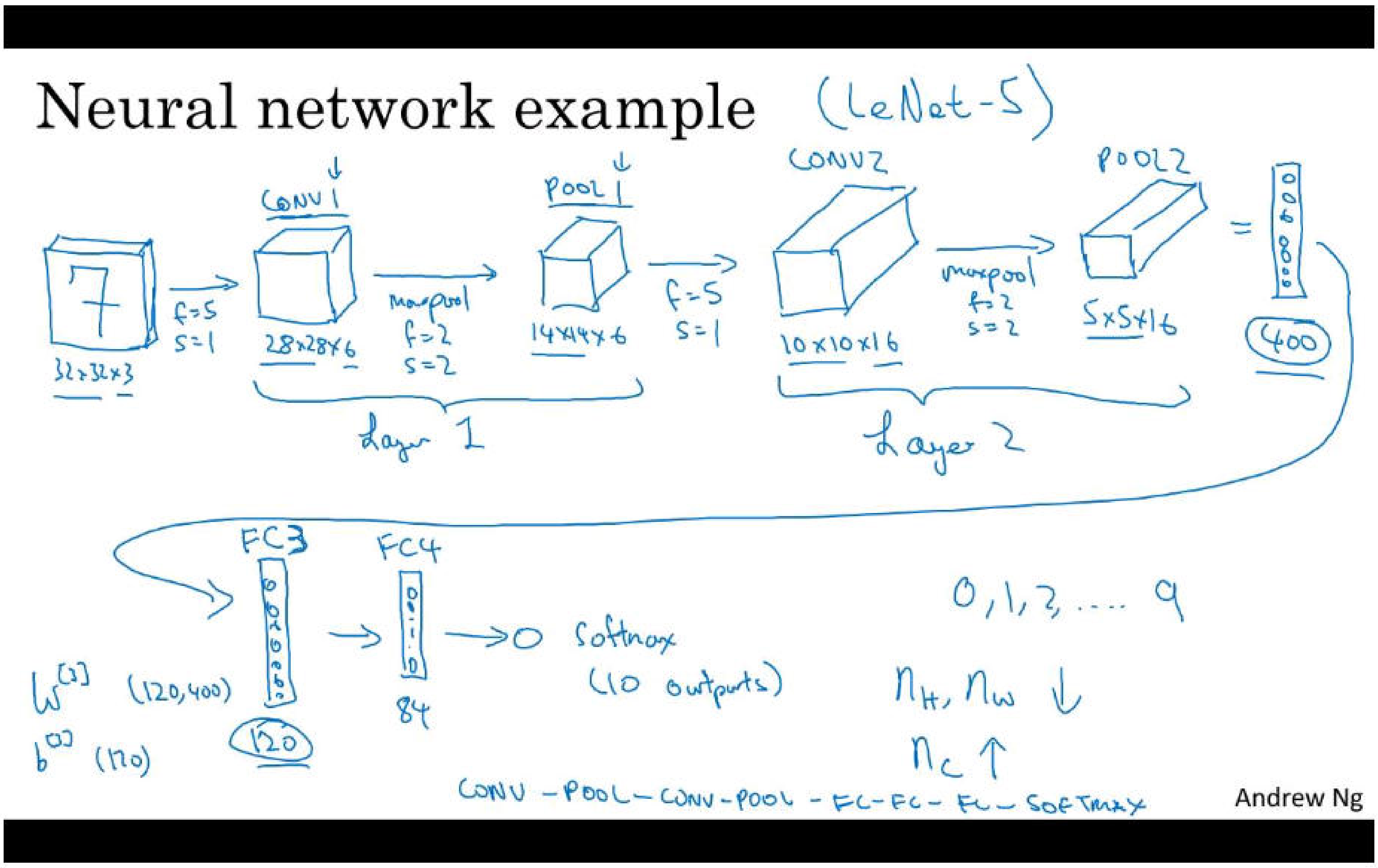
Pooling Layer.

2个超参, f, s, 常用: f=2, s=2; f=3, s=2.

- Max Pooling
- Average Pooling.

对每个channel都做.

典型 CNN



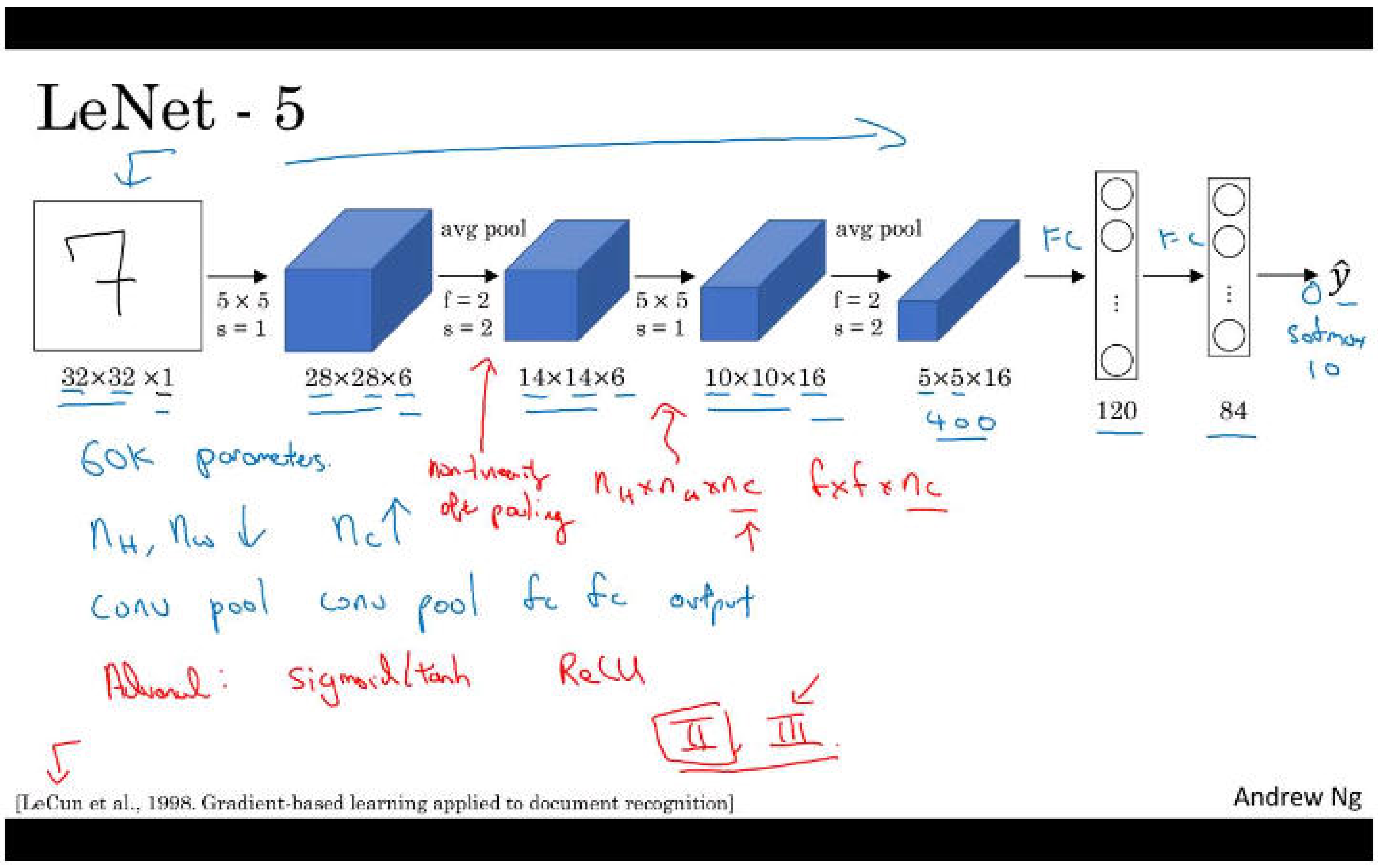
size↓, 一个 w, b 算一层.

卷积的优势:
- 参数共享: 特征提取复用
- 稀疏连接:

translation invariance.

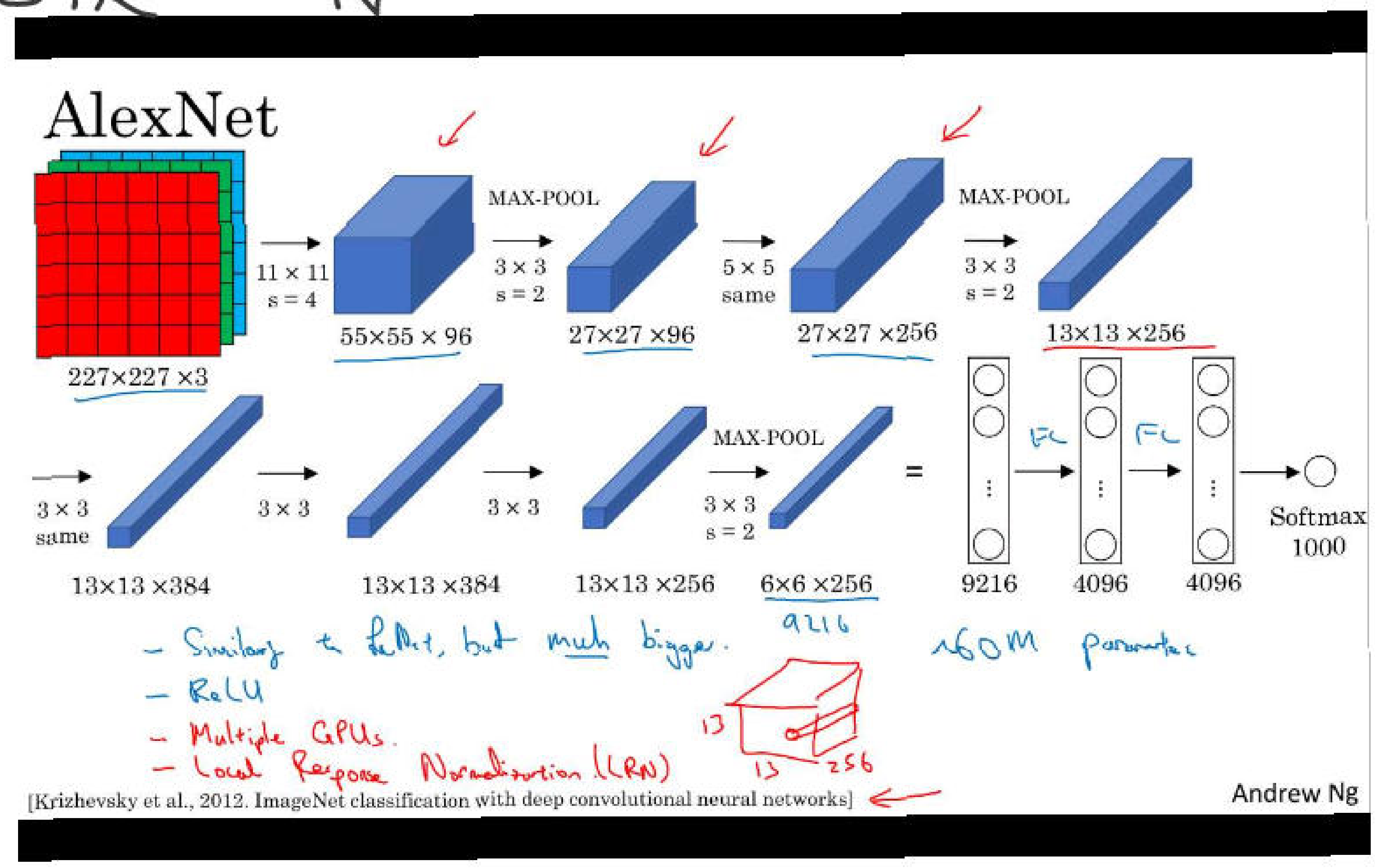
1 12

Classic CNN.



$n_H, n_W \downarrow$. 用 Sigmoid / tanh 激活, Pool 也加了激活.

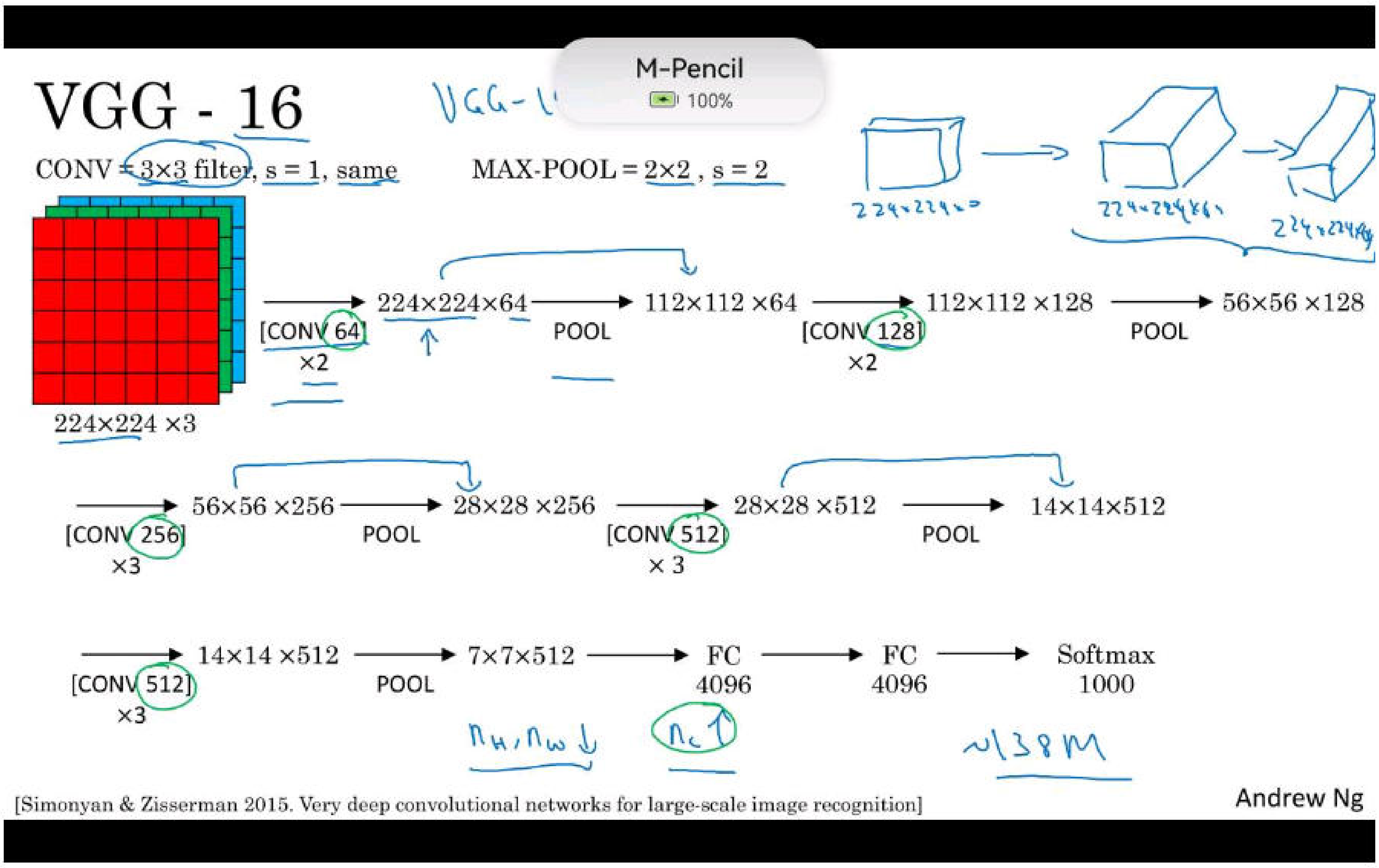
参数规模: 60k



Act: Relu, 参数规模: 60M.

filter 数量大大增加.

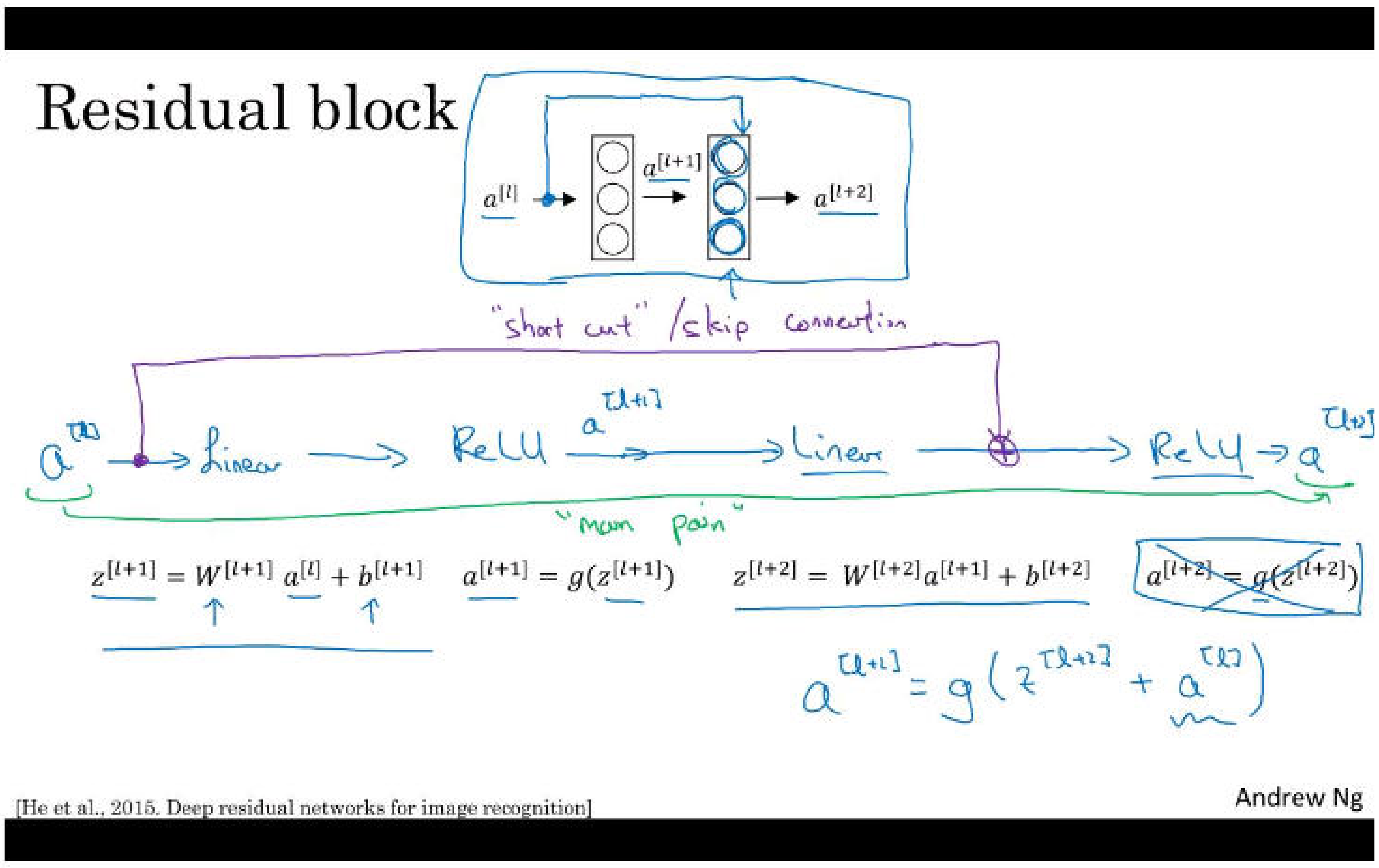
1 12



S=2, size 减半. # filter 加倍. 1000 类 SoftMax.

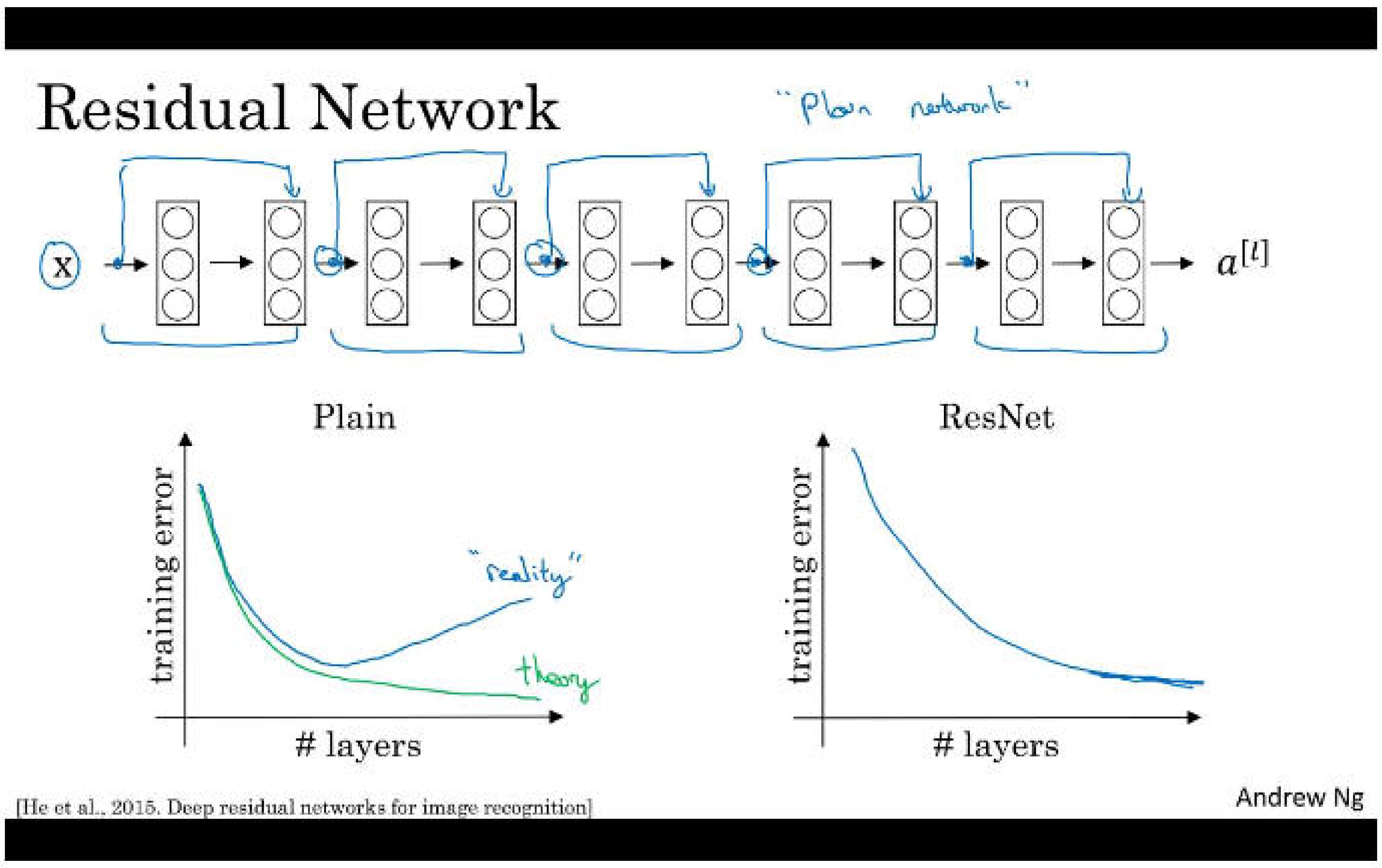
参数规模：138 M.

残差网络.



$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$. "short-cut".

1 12



在训练集上的 Error 即使 depth↑，也是↓.

为什么好用？

1): Reg 后， $a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$ (Relu).

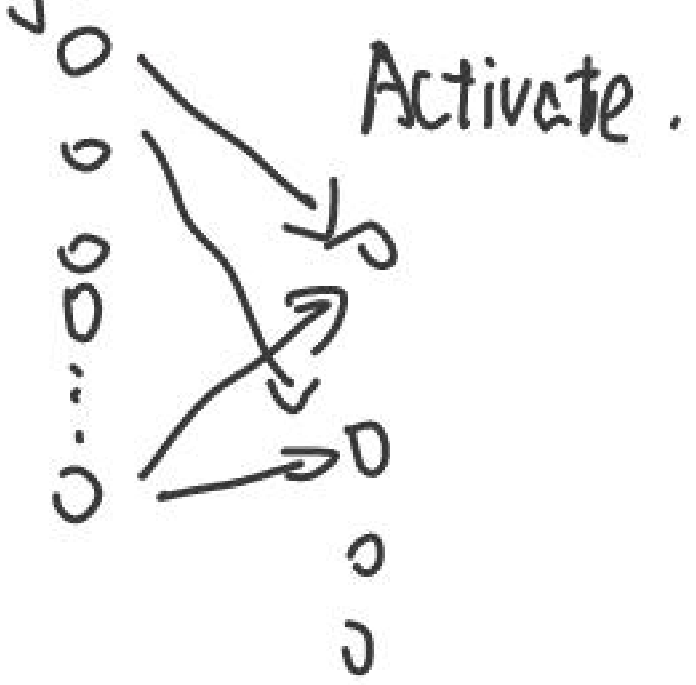
$$= g(\underbrace{w^{[l+2]}}_{\approx 0} a^{[l+1]} + \underbrace{b^{[l+2]}}_{\approx 0} + a^{[l]}) = a^{[l]}$$

学会了 I 关系.

2): 若 $a^{[l+2]}$ 与 $a^{[l]}$ 维度不同， $W_s a^{[l]}$ 调整维度. W_s 可以 0 pad $a^{[l]}$，也可自学习.

1×1 卷积.

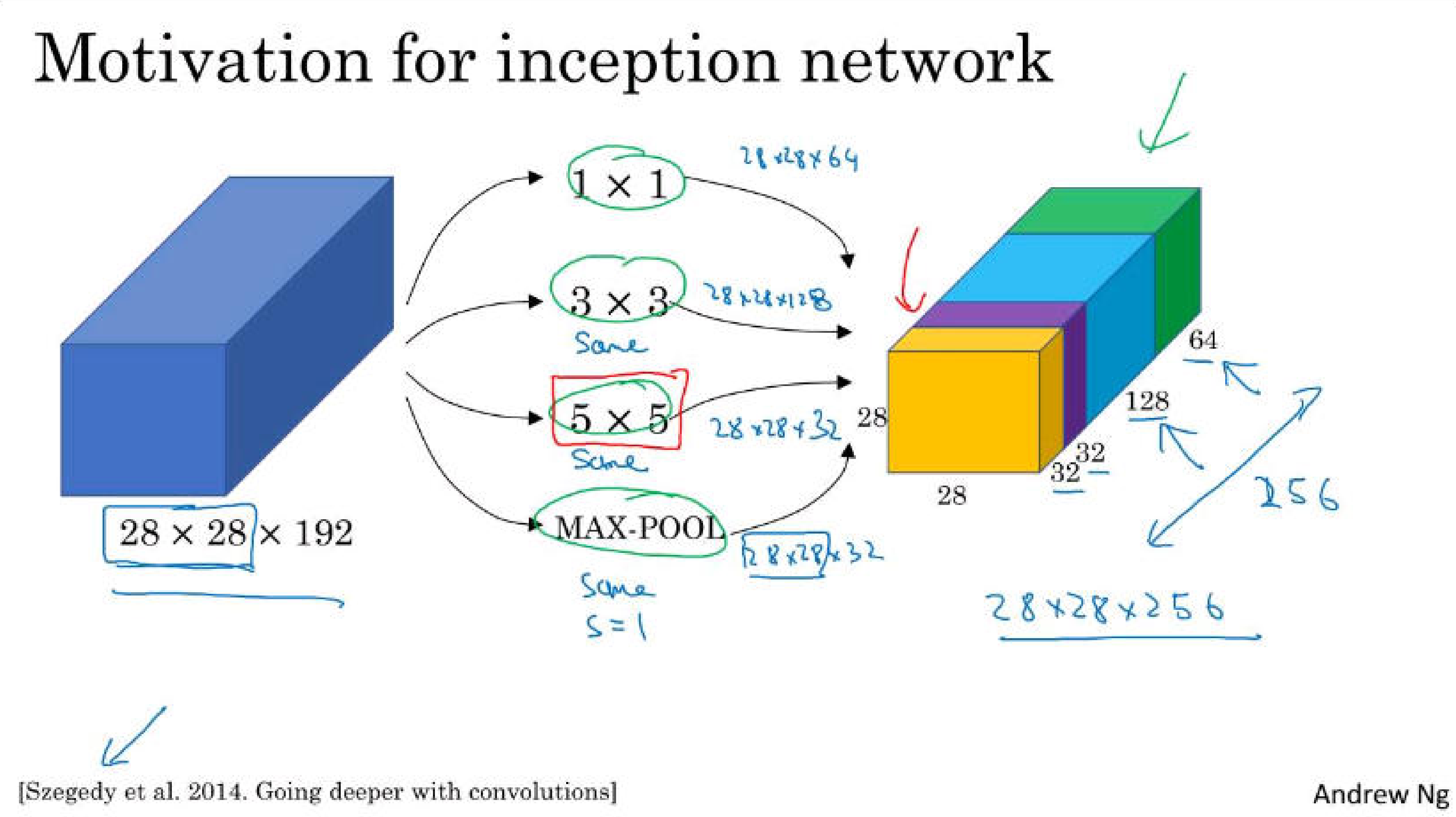
类似于:



相当于一层 FL.

网络中的网络.

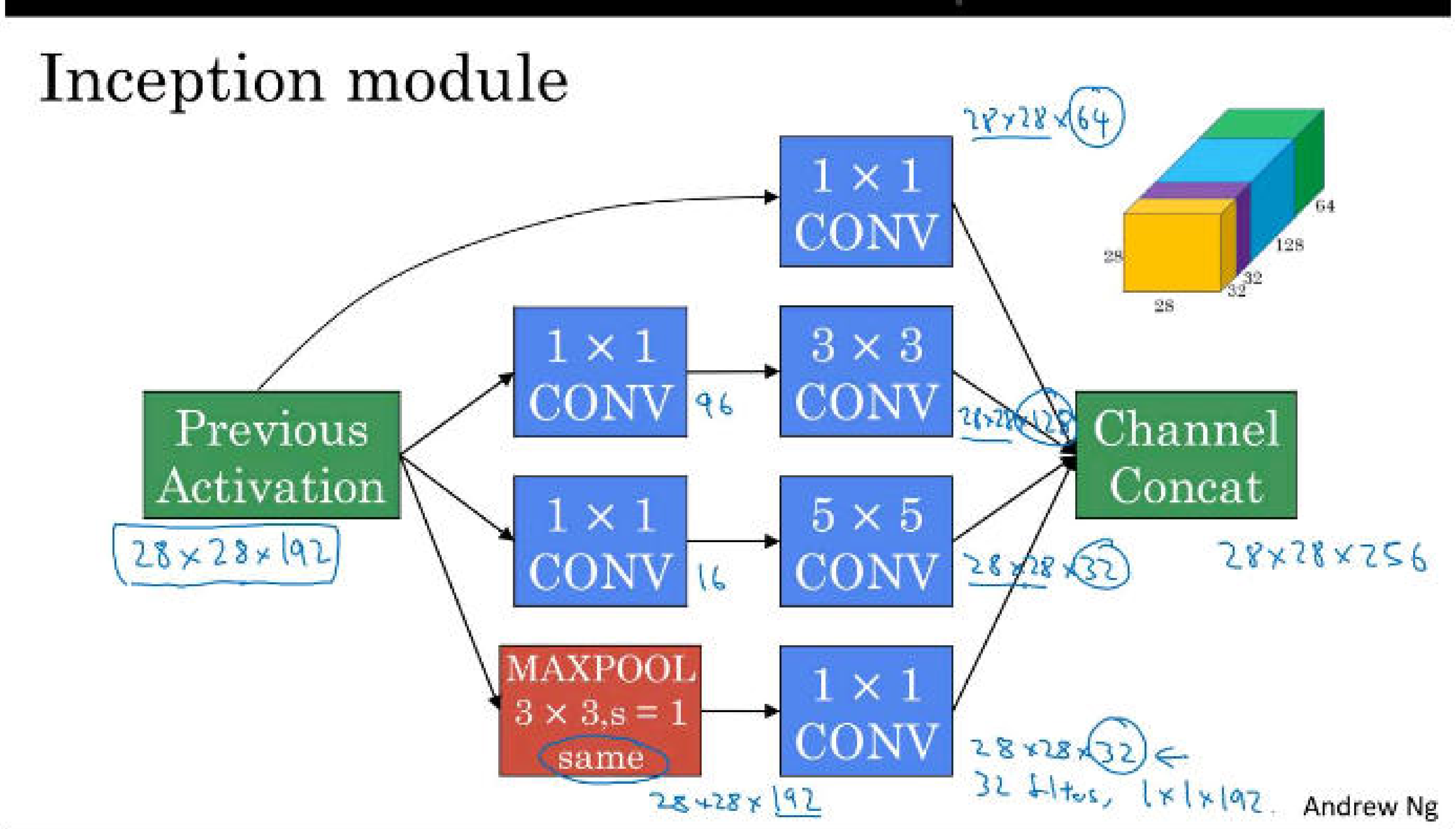
1 12



inception网络集成了各种filter，自动选择使用哪一个filter.

缺点：计算代价太高.

利用1×1 filter可以降低Computational Cost.



Inception网络由基本的module变形、堆叠而成.

1 12

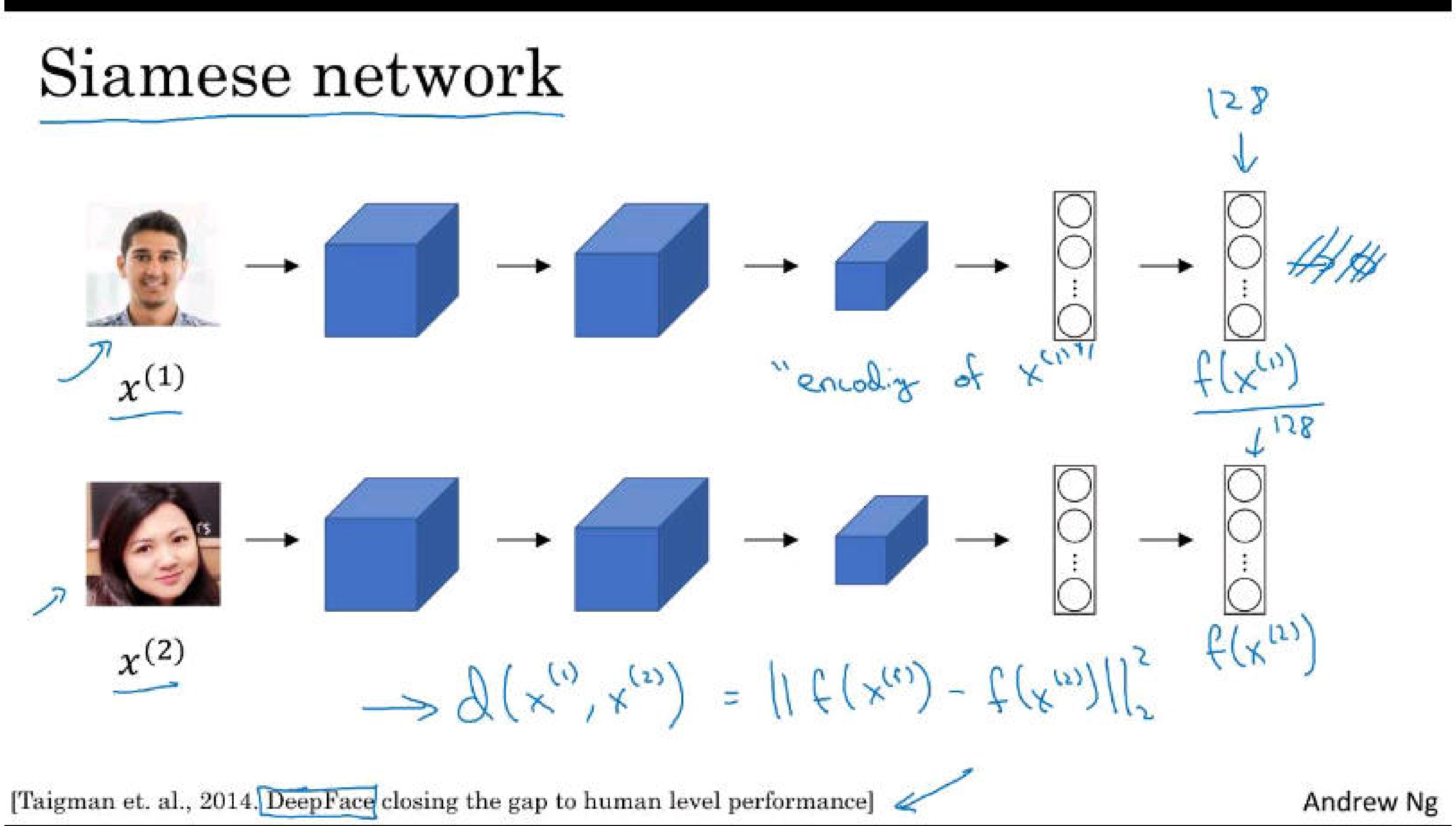
Transfer Learning.

只训练 SoftMax (少量 DataSet).
训练最后几层 (中等 Data Set).
重新训练 (用预训练参数 Initialize). (大量一).

Data Augmentation.

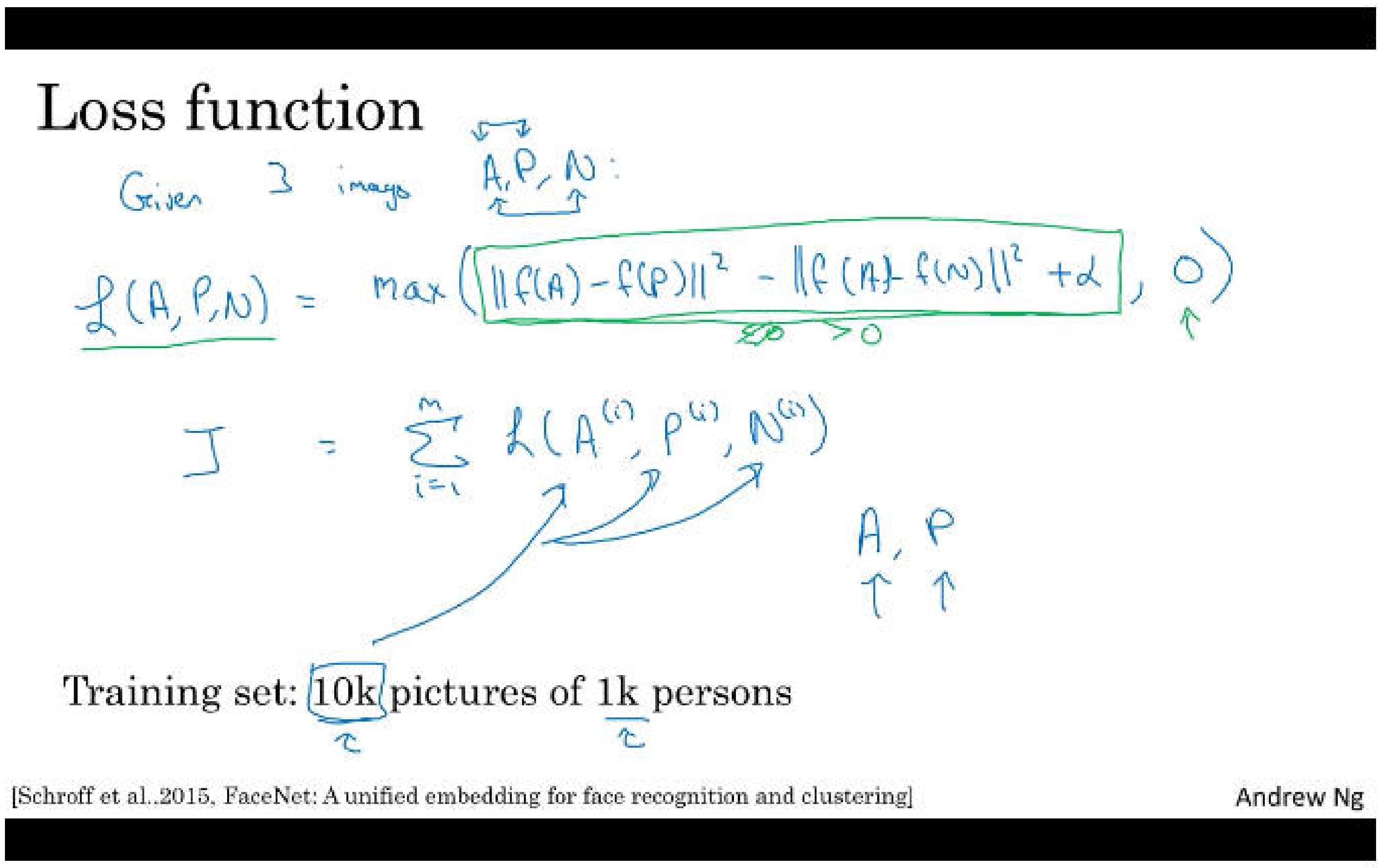
图像变换: Mirrior、Shearing …
色彩变换: Color Shifting.

人脸识别的 CNN



是一种 Encoder. 用于 d 的计算.

1 13



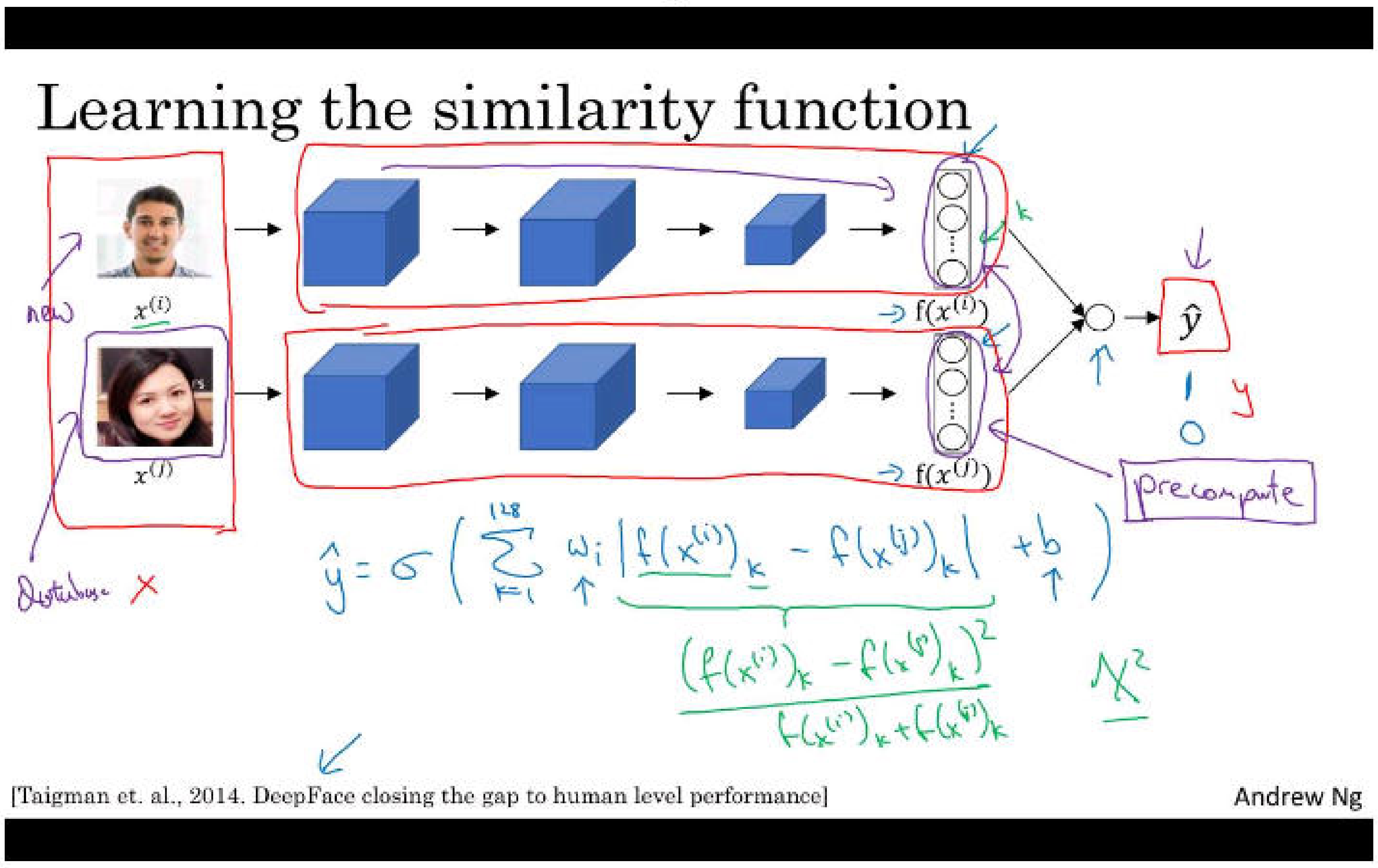
Siamese 的 Loss.

$$L(A,P,N) = \max\{\|f(A)-f(P)\|^2 - \|f(A)-f(N)\|^2 + \alpha, 0\}.$$

$$\|f(A)-f(P)\|^2 + \alpha \le \|f(A)-f(N)\|^2.$$

$$J(\theta) = \sum_{i=1}^{m} L(A^{(i)}, P^{(i)}, N^{(i)})$$

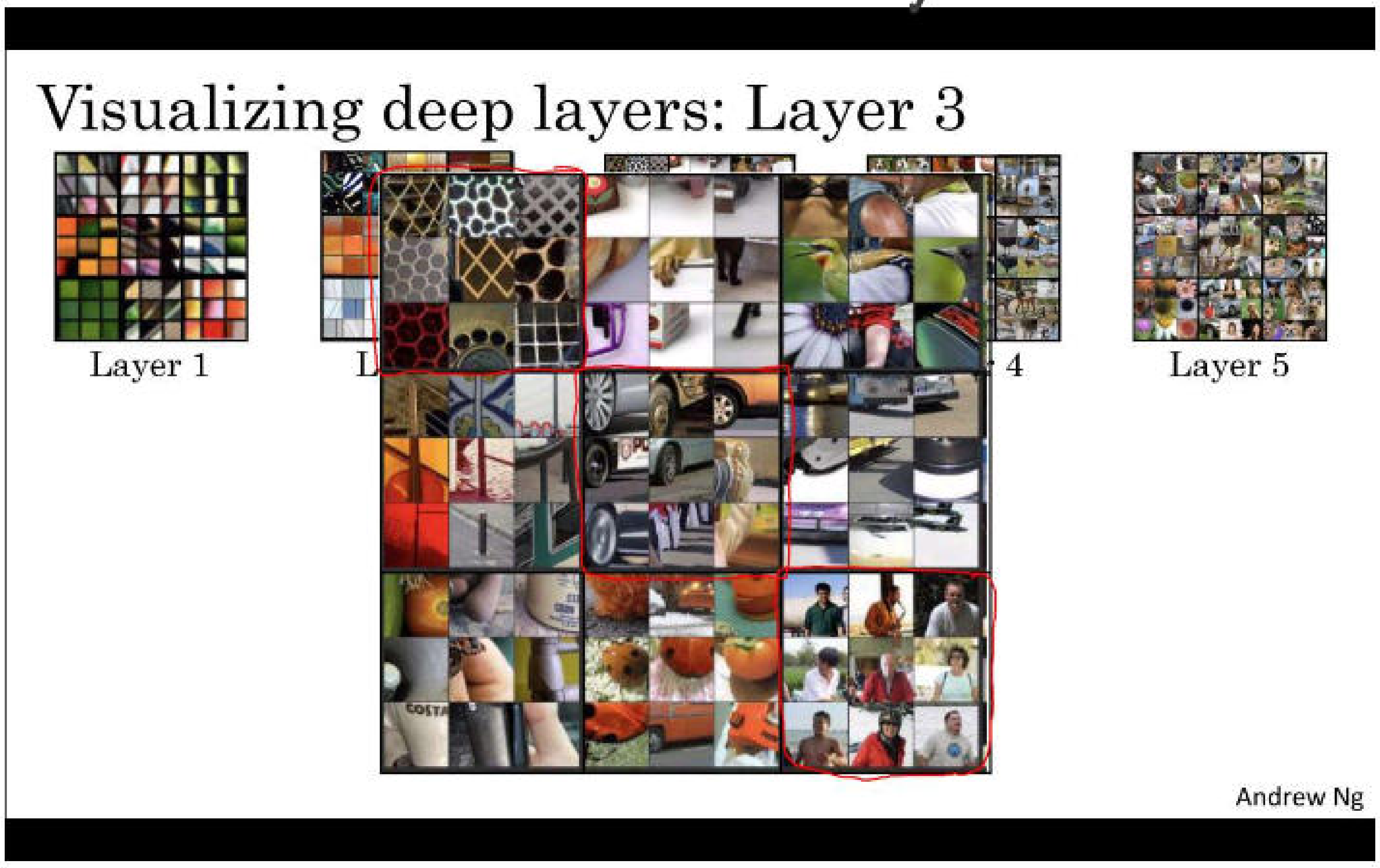
人脸识别的二分类.

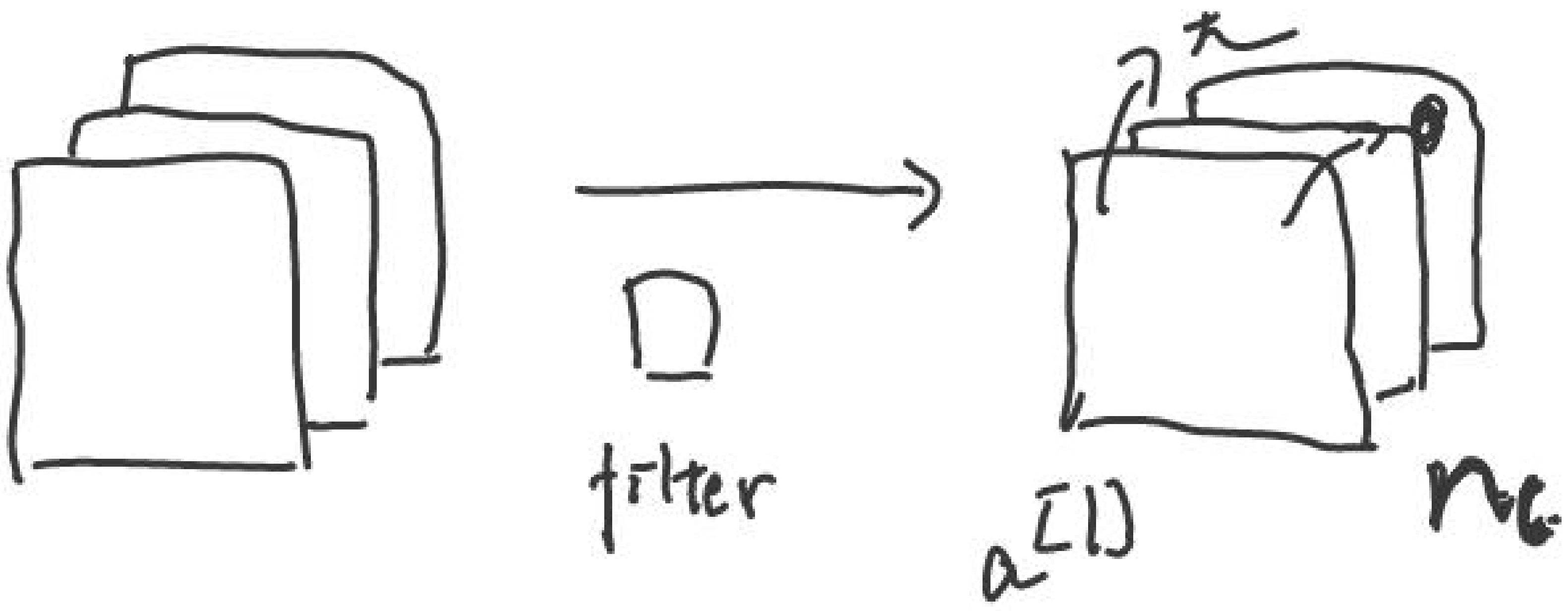


多一层 Logistic 回归，训练一个分类器.

1 13

Neural Transfer.





卷积次数↑，提取的模式与特征越复杂。

Content cost funct

Content: C

Style: S

Generation: G

$$J(G) = \alpha J_{content}(C, G) + \beta J_{style}(S, G)$$

- Say you use hidden layer l to compute content cost.
- Use pre-trained ConvNet. (E.g., VGG network)
- Let $a^{[l](C)}$ and $a^{[l](G)}$ be the activation of layer l on the images
- If $a^{[l](C)}$ and $a^{[l](G)}$ are similar, both images have similar content

$$J_{content}(C,G) = \frac{1}{2}\|a^{[l](C)} - a^{[l](G)}\|^2$$

[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng

$$J_{content} = \|a^{[l](C)} - a^{[l](G)}\|_F^2 .$$

1 13

Style matrix

Let $a^{[l]}_{i,j,k}$ = activation at (i, j, k). $G^{[l]}$ is $n_c^{[l]} \times n_c^{[l]}$

H W C

n_c

$G^{[l]}_{kk'}$

$k = 1, \dots, n_c^{[l]}$

$$\rightarrow G^{[l](S)}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a^{[l](S)}_{ijk} \, a^{[l](S)}_{ijk'}$$

"Gram matrix"

$$\rightarrow G^{[l](G)}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a^{[l](G)}_{ijk} \, a^{[l](G)}_{ijk'}$$

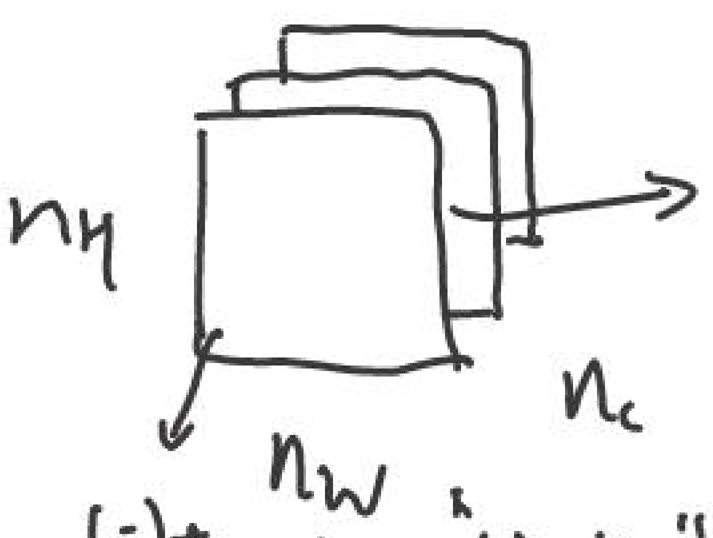
$$J^{[l]}_{style}(S, G) = \frac{1}{(\dots)} \| G^{[l](S)} - G^{[l](G)} \|_F^2$$

$$= \frac{1}{(2 n_H^{[l]} n_W^{[l]} n_C^{[l]})^2} \sum_k \sum_{k'} \left(G^{[l](S)}_{kk'} - G^{[l](G)}_{kk'} \right)^2$$

β

[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng



filter 2, "Color". Style 体现为不同特征组合.

filter 1. "纹理".

$$G^{[l]}_{kk'} = \sum_i \sum_j a^{[l]}_{ijk} a^{[l]}_{ijk'}$$ (展开为向量的内积)

$$J^{[l]}_{style}(S, G) = \frac{1}{K} \cdot \| G^{[l](S)} - G^{[l](G)} \|_F^2$$

$$K = (2 n_W^{[l]} n_H^{[l]} n_C^{[l]})^2.$$

$$J = \alpha \cdot J_{content} + \beta \cdot J_{style}.$$

$$G = G - \alpha \cdot \frac{\partial J}{\partial G}.$$